HRVATSKA POŠTANSKA BANKA (HPB)
PSD2 OPEN API - PRODUCTION

TPP IMPLEMENTATION GUIDE

Document version	3.2
Release date	2022-06-29

Change Log & Notes

Version ID	Release Date	Notes[1]
v3.2	2022-06-29	Updated **chapters**: 4.2 | 8. New **chapters**: *none* Deleted **chapters**: 4. Proofreading chapters: 1.4 | 1.5 | 2.4 | 5.2 GENERAL NOTES: ○ We now support new request header “TPP-Decoupled-Preferred” that can be used for choosing DECOUPLED flow. For more info and system behavior in combination with “TPP-Redirect-Preferred” request header, please refer to the updated chapter 4.2. ○ We have published a new document that regulates procedure for TPP when creating support request for Production environment. We advise all TPPs that use our Production environment to read and distribute it internally on a need-to-know basis.
v3.1	2022-03-25	Updated **chapters**: 3.1 | 4. New **chapters**: 1.4 | 1.5 | 2.5 | 5. | 6. | 7. Deleted **chapters**: 4.1 | 4.2 | 4.3 GENERAL NOTES: ○ This version has been updated with data on how to use “Decoupled” flow where PSD2 resource can be authorized via “push” notification on PSU mobile banking app. ○ Also, there has been provided more details on AIS resource rules (*frequencyPerDay, accessing transaction history and pagination*). ○ Regarding the PIS resource, there has been provided info on initiation of single payments without debtor account. IMPORTANT NOTICE: We would like to announce that HPB will stop supporting authorization using the here specified “simple” Redirect flow. As per RTS, this change will not happen within next 3 month (starting from the release of this document version) allowing all TPPs to reconsider their implementation processes. Exact date of this change will be additionally announced on HPB official website within PSD2 dedicated page: https://www.hpb.hr/hr/psd2-hpb-open-api-portal/318 TPP can continue to use oAuth Redirect flow or Decoupled flow instead.

[1] * *Updated chapters: It contains a significant change - newly inserted/deleted/updated info.*
* *New chapters: It contains data about new functionalities/limitations or other guidelines.*
* *Deleted chapters: When deleted, the specified chapter number refers to number it had in previous version.*
* *Proofreading chapters: No significant change has been made – only corrections to typo, grammar, and spelling mistakes or providing better explanations.*

v2.0	2020-12-31	Proofreading version.
v1.0	2019	Initial version.

Glossary / List of Abbreviations and Terms Used in this Document

Abbreviation / Term	Expansion / Description
TPP	Third party provider -> A legal entity acting either as AISP, PISP, PIISP.
AIS	Account Information Services
PIS	Payment Initiation Services
PSU/User	Individual (person) that represents itself solely or individual granted with certain banking services on behalf of corporate client (business entity).
PSU-ID	Represents OIB -> national personal identification number of natural person (11 digits).
IAM	Identity and Access Management system
CA	Certification Authority
OAuth 2.0	Industry-standard protocol for authorization
SCA	Strong Customer Authentication
CoF/FCS	Confirmation of Funds/Funds Check Service
Redirect flow	SCA approach where PSU uses PSD dedicated system (X2A interface) for authorization.
Decoupled flow	SCA approach where PSU uses HPB mobile banking app/token (mHPB) for authorization.

Contents

1. Introduction

This document is used to describe the functionalities of the HPB PSD2 Open API in Production environment.

1.1. Purpose

Purpose of this document is to provide details on how to use the product from a functionality point of view.

1.2. Intended audience

Main audience of this document are TPPs that want to register their application at bank in order to use PSD2 API methods exposed by the bank.

1.3. Scope

Descriptions in this document describe the following processes and flows:

1. TPP application registration
2. Creating and authorizing resources
3. Using different methods of authorization

1.4. General instructions for implementation

Hrvatska Poštanska Banka d.d. (hereinafter: **HPB**) is a member of Croatian Banking Association (hereinafter: HUB) and follows HR-Specific implementation guidelines – National PSD2 framework. For more details on that, TPP is advised to check website of HUB: https://www.hub.hr/en/psd2-open-api

For HPB PSD2 Open API, TPP should be aware that there are certain headers that should be included in every request in order to be successful:

- **PSU-ID**[2]**:** OIB value (*mandatory header, variable, see Glossary*),
- **PSU-ID-Type:** “customerNumber” (*mandatory header, fixed value*),
- **X-Request-ID:** GUID/UUID value (*mandatory header, variable, generated by TPP*).

TPP should regularly check HPB PSD2 dedicated webpage that contains our official documentation and announcements: https://www.hpb.hr/hr/psd2-hpb-open-api-portal/318

Open API base URL: https://api.openbanking.hpb.hr (hereinafter: “{{ApiBaseUrl}}”)

IAM base URL: https://iam.openbanking.hpb.hr (hereinafter: “{{IamBaseUrl}}”)

Berlin Group-Implementation Guide: Please note that HPB PSD2 Open API is based on v1.3.9 of BG-IG.

Refresh token: Not supported.

[2] For corporate clients, it represents the PSU-ID of the authorized natural person that must be registered at the Bank’s system and must have any of the online channels active. PSU-ID (OIB) of corporate client is not used at all.

1.5. Supported AIS options and PIS products

HPB supports the following products:

- Account Information Service (consents)
    - Dedicated consent
    - Global consent
    - Bank-Offered consent
- Payment Initiation Service (single and bulk payments)
    - payments/domestic-credit-transfers-hr (HRK only)
    - payments/instant-domestic-credit-transfers-hr (HRK only)
    - payments/sepa-credit-transfers (EUR only)
    - payments/cross-border-credit-transfers*
    - payments/hr-rtgs-payments
    - payments/target-2-payments*
    - bulk-payments/pain.001-credit-transfers* (HRK only)
- Signing Basket
    - Supports grouping of same-type single payments only

* Available to Corporate clients only (business entity).

2. Registration

2.1. Goal

Register your application in order to gain access to PSD2 API exposed by the bank.

2.2. Preconditions

Obtained valid X509 Certificate from trusted CA that satisfies requirements stated in ETSI TS 119 495 V1.2.1 directive.

Installed Postman or similar application.

2.3. How to access

Through the API call from TPP application, Postman client or similar apps.

2.4. Overview

In order to use PSD2 services exposed by the bank, TPP needs to make a request to the specific endpoint in order to register itself and to get credentials that are needed for OAuth2 SCA. Endpoint that is used for TPP application registration is: POST *{{IamBaseUrl}}/connect/register.*

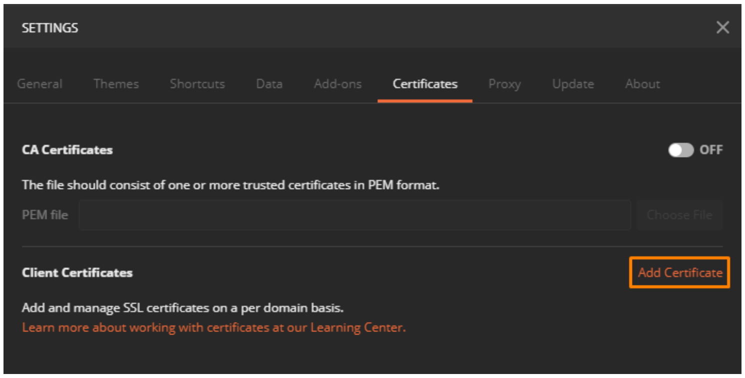
The payload of this request must be in JSON format and must contain following fields:

- Redirect URIs (*redirect_uris*)
  Required, list of URIs that TPP wants to register for redirection after successful completion of OAuth2 flow
- Post Logout Redirect URIs *(post_logout_redirect_uris)*
  Optional, list of URIs that TPP wants to register for redirection after user logs out from the IAM application
- Logo URI *(logo_uri)*
  Optional, URI to client logo
- Front Channel Logout URI *(front_channel_logout_uri)*
  Specifies logout URI at client for HTTP based front-channel logout
- Back Channel Logout URI *(back_channel_logout_uri)*
  Specifies logout URI at client for HTTP based back-channel logout
- Client URI *(client_uri)*
  Optional, URI to further information about TPP

Example payload:

```
{
  "post_logout_redirect_uris": [
    "https://www.example.com/oauth2/callback"
  ],
  "client_uri": "https://www.example.com/app",
  "logo_uri": "https://www.example.com/logo.svg",
  "redirect_uris": [
    "https://www.example.com/oauth2/callback"
  ]
}
```

In order to successfully perform Mutual TLS with the IAM application, TPP needs to provide X509 Certificate for authentication and to sign requests using private key that is associated with the public key from used certificates. To achieve this in Postman go to **File->Settings**. In new window click on **Certificates** tab. There is a button called **Add Certificate** under **Certificates** tab.



CA Certificates

The file should consist of one or more trusted certificates in PEM format.

Client Certificates

Add and manage SSL certificates on a per domain basis.
Learn more about working with certificates at our Learning Center.

Figure 1: Adding certificate for Mutual TLS

Clicking on this button will open new window. In this window you need to fill in following fields:

- Host
  Required, base path to the IAM application
- CRT file
  Path to the file that contains X509 Certificate in PEM format
- KEY file
  Path to the file that contains Private Key in PEM format
- PFX file
  Path to the file that contains both X509 Certificate and Private Key in PFX format
- Passphrase
  Passphrase for opening PFX file

TPPs that have CRT and KEY files should not use **PFX file** and **Passphrase** fields, also, TPPs that have certificate in **PFX** format should not use **CRT file** and **KEY file** fields.

Upon completion, TPP should send a registration request: POST *{{IamBaseUrl}}/connect/register*. If the request was successful, TPP will get a response that looks similar to this example:

```
{
  "client_id": "63.certificate",
  "client_secret": "Certificate thumbprint",
  "client_name": "63 Certificate Client",
  "grant_types": "authorization_code,password,client_credentials",
  "scope": "PSD2 PIS:<paymentId> AIS:<consentId>",
  "client_uri": "https://www.example.com/app",
  "logo_uri": "https://www.example.com/logo.svg",
  "redirect_uris": [
    "https://www.example.com/oauth2/callback"
  ],
  "post_logout_redirect_uris": [
    "https://www.example.com/oauth2/callback"
  ],
  "front_channel_logout_uri": null,
  "back_channel_logout_uri": null
}
```

This response contains data that will be needed later for starting the OAuth2 flow for authorizing AIS and PIS resources. Response contains following fields:

- Client Id
  Id of client that was created for TPP during registration
- Client Secret
  Secret for the created client. If this field has value "Certificate Thumbprint" that means that secret for the created client is thumbprint from certificate that was used for TPP registration
- Client Name
  Friendly client name
- Grant Types
  Allowed grant types
- Scope
  Allowed scopes
- Client URI
- Logo URI
- Redirect URIs
- Post Logout Redirect URIs
- Front Channel Logout URI
- Back Channel Logout URI

2.5. Certificate management

If TPP certificate for PSD2 services has been changed, the following should be done:

- **Renewed certificate**: If TPP certificate has expired, TPP should renew it and then make a new request with a new certificate, as described previously, to an endpoint: POST *{{IamBaseUrl}}/connect/register.* If a new certificate is attached and valid and if the Issuer and Subject match to previously registered one, TPP certificate will be updated in our system.
- **New certificate**: If the Issuer or Subject of the certificate has changed, you should complete full registration, as previously described. It will be considered as a new *client_id*.

3. AIS and PIS resources using the OAuth2 flow

3.1. Goal

Goal of this section is to successfully create AIS resource (consent) and PIS resource (payment), to start authorization for created resources and to authorize resources using the OAuth2 protocol.

In order to use this method of authentication, TPP should add extra headers to the POST request when creating resource, specified here:

- TPP-Redirect-Preferred: **true** *(optional, recommended)*
- TPP-Redirect-URI: https://example.com/OK *(mandated for oAuth flow)*
- TPP-Nok-Redirect-URI: https://example.com/NOK *(optional)*

3.2. Creating and authorizing AIS resources

3.2.1. Creating AIS consent

In order to read account details/transactions/balances (*depending on scope of consent*), TPP needs to get consent from user. First step in doing this is creation of consent resource. To do this TPP has to make call to *POST /v1/*consents endpoint. Request should have payload that is similar to this (*for full description of payload and headers refer to Berlin Group NextGen PSD2 Documentation*):

Request example:

```
{
  "access": {
       "availableAccounts": "allAccounts"
  },
  "recurringIndicator": true,
  "validUntil": "YYYY-MM-DD",
  "frequencyPerDay": 4,
  "combinedServiceIndicator": false
}
```

As in guide for TPP application registration, TPP should add certificate that will be used for Mutual TLS.

3.2.2. Starting authorization for AIS consent

When consent resource is successfully created, TPP has to make call to: *POST /v1/consents/{consentId}/authorizations* endpoint where *consentId* is id of the consent that was previously created. If the authorization was successfully created, response payload should be similar to this:

```
{
    "scaStatus": "received",
    "authorizationId": "9e52ecef8e8044dfadaebe430ce0494f",
    "scaMethods": [
        {
            "authenticationVersion": "1.00",
            "authenticationMethodId": "SCA Method 3",
            "name": "SCA Method 3"
        }
    ],
    "_links": {
        "scaOAuth": {
            "href": "https://iam.sandbox.openbanking.hpb.hr/"
        },
        "scaStatus": {
            "href": "v1/consents/599c4cbb-34f2-4b01-820d-eb0d86dffa15/authorisati-
ons/9e52ecef8e8044dfadaebe430ce0494f"
        },
        "confirmation": {
            "href": "v1/consents/599c4cbb-34f2-4b01-820d-eb0d86dffa15/authorisati-
ons/9e52ecef8e8044dfadaebe430ce0494f"
        }
    }
}
```

3.2.3. AIS consent authorization using OAuth2 protocol

For authorizing Consent using OAuth2 flow, TPP needs URL from *scaOAuth* field. To start OAuth2 protocol, TPP has to redirect client from its application to IAM application. Endpoint on which user needs to be redirected is *scaOAuth/connect/authorize,* where *scaOAuth* is value of corresponding field in the response of the request for starting the authorization. User should be redirected with following query parameters:

- **Grant Type** (grant_type) **–** This field needs be equal to *code*
- **Response Type** (response_type) – This field needs to be equal to *code*
- **Redirect URL** (redirect_uri) **–** URL on which TPP wants user to be redirected after finishing SCA, should be equal to some of URLs that are provided on TPP application registration under the *Redirect URIs* field
- **Client ID** (client_id)**–** Id of client that was created for TPP on TPP registration
- **Scope** (scope) **–** Scope should have value that equals to AIS:<consentId> where <consentId> should be replaced with id of the consent that we want user to authorize

Redirect URL should be in format similar to this:

https://iam.sandbox.openbanking.hpb.hr/connect/authorize?client_id=client.id&scope=AIS:e3bf80a0-996e-47e5-8840-b3b83eaa29ed&redirect_uri=https://www.redirect.com/oauth-callback&grant_type=code&response_type=code

If user has done authentication successfully, user will be redirected to the URI that TPP provided in *redirect_uri* field with following parameters in string format:

- Code (code) – one-time code that will be used for obtaining access token by TPP
- Scope (scope) – scopes that were granted
- Session State (sessionState) – this field can be omitted

TPP should make a request for access token in this callback method. This access token will be used as authorization data that is required for consent authorization. To obtain access token, TPP has to send a request to POST *scaOAuth/connect/token/mtls,* with content type *application/x-www-form-urlencoded* and following parameters in request body:

- Client ID (client_id) - Id of client that was created for TPP on TPP registration
- Scope (scope) – this field should be equal to the scope parameter received in callback
- Code (code) – code that was received in callback
- Redirect URI (redirect_uri) – redirect URI that was used I */connect/authorize* request
- Grant Type (grant_type) – This field needs to be equal to "authorization_code"

As in guide for TPP application registration, TPP should add certificate that will be used for Mutual TLS. Response body of the successful request will contain access token.

Response example:

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

3.2.4. Finishing authorization

In order to finish consent authorization, TPP needs to send a request to PUT */v1/consents/{consentId}/authorisations/{authorisationId}*. Consent ID is id of consent that is authorizing, and authorizationId is id of authorization resource that was created for consent authorization.

Request body of this field has to be in application/json format and must contain field “scaAuthenticationData”. Value of this field has to be equal to the access token that was obtained through the OAuth2 protocol.

Request body example:

```
{
  "scaAuthenticationData":
"eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2OTJGQzc2RkI1Qjk4OTBCQUFGQ0JCQUYyOTQyNDZDQjUzMzdEMEYiL
CJ0eXAiOiJKV1QiLCJ4NXQiOiJWcEw4ZHZ0Ym1KQzZyOHU2OHBRa2JMVXpmUTgifQ.eyJuYmYiOjE2NDY5Mjgz
ODAsImV4cCI6MTY0NjkzMTk4MCwiaXNzIjoiaHR0cHM6Ly9pYW0udGVzdC5vcGVuYmFua2luZy5ocGIuaHIiLC
JhdWQiOlsiaHR0cHM6Ly9pYW0udGVzdC5vcGVuYmFua2luZy5ocGIuaHIvcmVzb3VyY2VzIiwiQXNzZWNvU0VF
LlBTRDIiXSwiY2xpZW50X2lkIjoiMjUudnJlYmFjLnRwcC1zdXBldXMiLCJzdWIiOiI0ODM4OGJjMi1iMWVmLT
Q5NjAtYTBiMC01MDk5Yzk0OWNhYzYiLCJhdXRoX3RpbWUiOjE2NDY5MjgzNzUsImlkcCI6ImxvY2FsIiwicHJv
dmlkZXIiOiIyNSIsImJhbmtfaWQiOiIxIiwiY3VzdG9tZXJfbnVtYmVyIjoiMzk1NTQ3NDM5MzEiLCJ1c2VyX2
5hbWUiOiIzOTU1NDc0MzkzMSIsInVzZXItdHlwZSI6InVzZXIiLCJwYXJ0eS1raW5kIjoiaW5kaXZpZHVhbCIs
ImUtYmFua2luZyI6ImUtYmFua2luZy1ub25lIiwiaXBfYWRkcmVzcyI6IjE3Mi4yNy4wLjk4Iiwic2NvcGUiOl
siQUlTOjljMmJkNmE5LTlkNzAtNDE5My04MDYzLWRmOTQyYjBiZGZlZCJdLCJhbXIiOlsicHdkIl19.X-
E9Oia7r2ds8b4NiRIYtZeBMkoMnqorTHlzGe9dgDENmK7Bh4SBjF4fxWiqgiXmdSFesgx-
SnHWgbL5C1v_A10WiQ9Ghh-lU0OjJAchX32YtRlQpaakux3Ew1PyTtCA-
xVHorc_2rI3o0Yd9aUSffUA4BqUxkT7sMHphMi_kyPRFxyuBvSFRTHHeF3D2Z3jDFKuVJVXnEsa6GjV1mv3_Ek
Fp5huq103fRt_2kpdY4d4w1gI1tm-
74idKwYRM2pVUZ0rTsNFFLlrkXPYUEAMUeJpzUd9MrooOTrJkhAy3YI2Ks8AEKydvljtLVMYJyRo1RRjsWzzfe
_Y60ZQaRuDVg"
}
```

3.3. Creating and authorizing PIS resources

3.3.1. Creating payment resource

To create payment TPP has to make call to *POST /v1/{payment-service}/{payment-product}* endpoint.

Request example (for full description of payload and headers refer to Berlin Group NextGen PSD2 Documentation):

```
{
  "debtorAccount": {
   "iban": "HR1223900010000000000",
   "currency": "HRK"
  },
  "instructedAmount": {
    "currency": "HRK",
    "amount": "11.26"
  },
  "creditorAccount": {
    "iban": "HR5423400091111111111"
  },
  "creditorName": "John Doe",
  "creditorAddress": {
        "streetName": "Test street",
        "buildingNumber": "1",
        "townName": "ZAGREB",
        "postCode": "10000",
        "country": "HR"
   },
  "remittanceInformationUnstructured": "Free description here",
  "requestedExecutionDate": "YYYY-MM-DD"
}
```

As in almost all previous requests, TPP should add certificate that will be used for Mutual TLS.

3.3.2. Starting authorization for payment resource

When payment resource is successfully created, TPP has to make call to: POST */v1/{payment-service}/{payment-product}/{paymentId}/authorisations* endpoint where *paymentId* is id of the payment resource that was previously created. If the authorization was successfully created, response payload should be similar to this:

```
{
    "scaStatus": "received",
    "authorizationId": "c7cce6816f404d8c977d96b3627db142",
    "scaMethods": null,
    "_links": {
        "scaOAuth": {
            "href": "https://iam.sandbox.openbanking.hpb.hr/"
        },
        "scaStatus": {
            "href": "v1/payments/domestic-credit-transfers-
hr/c2e2de8d641f471dbf4b17eb36d89fbe/authorisations/c7cce6816f404d8c977d96b3627db142"
        },
        "confirmation": {
            "href": "v1/payments/domestic-credit-transfers-
hr/c2e2de8d641f471dbf4b17eb36d89fbe/authorisations/c7cce6816f404d8c977d96b3627db142"
        }
    }
}
```

3.3.3. Payment resource authorization using OAuth2 protocol

For authorizing payment resource using OAuth2 flow, TPP needs URL from *scaOAuth field.* To start OAuth2 protocol, TPP needs to redirect client from its application to IAM application. Endpoint on which user needs to be redirected is *scaOAuth/connect/authorize,* where *scaOAuth* is value of corresponding field in the response of the request for starting the authorization. User should be redirected with following query parameters:

- Grant Type (grant_type) – This field needs be equal to *code*
- Response Type (response_type) – This field needs to be equal to *code*
- Redirect URL (redirect_uri) – URL on which TPP wants user to be redirected after finishing SCA, should be equal to some of URLs that are provided on TPP application registration under the *Redirect URIs* field
- Client ID (client_id)– Id of client that was created for TPP on TPP registration
- Scope (scope) – Scope should have value that equals to PIS:<paymentId> where <paymentId> should be replaced with id of the payment resource that we want for user to authorize

Redirect URL should be in format similar to this:

*https://iam.sandbox.
openbanking.hpb.hr/connect/authorize?client_id=id.client&scope=PIS:783867aab4bc439291c6c5e2e6b3db6f&redirect_uri=https://www.returnurl.com/oauthcallback&grant_type=code&response_type=code*

If user has done authentication successfully, user will be redirected to the URI that TPP provided in *redirect_uri* field with following parameters in string format:

- Code (code) – one-time code that will be used for obtaining access token by TPP
- Scope (scope) – scopes that were granted
- Session State (sessionState) – this field can be omitted

TPP should make a request for access token in this callback method. This access token will be used as authorization data that is required for payment resource authorization. To obtain access token, TPP has to send a request to POST *scaOAuth/connect/token/mtls,* with content type *application/x-www-form-urlencoded* and following parameters in request body:

- Client ID (client_id) - Id of client that was created for TPP on TPP registration
- Scope (scope) – this field should be equal to the scope parameter received in callback
- Code (code) – code that was received in callback
- Redirect URI (redirect_uri) – redirect URI that was used I */connect/authorize* request
- Grant Type (grant_type) – This field needs to be equal to authorization_code

As in guide for TPP application registration, TPP should add certificate that will be used for Mutual TLS. Response body of the successful request will contain access token.

Response example:

```
{
  "access_token":
"eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2OTJGQzc2RkI1Qjk4OTBCQUFGQ0JCQUYyOTQyNDZDQjUzMzdEMEYiL
CJ0eXAiOiJKV1QiLCJ4NXQiOiJWcEw4ZHZ0Ym1KQzZyOHU2OHBRa2JMVXpmUTgifQ.eyJuYmYiOjE2NDY5Mjgz
ODAsImV4cCI6MTY0NjkzMTk4MCwiaXNzIjoiaHR0cHM6Ly9pYW0udGVzdC5vcGVuYmFua2luZy5ocGIuaHIiLC
JhdWQiOlsiaHR0cHM6Ly9pYW0udGVzdC5vcGVuYmFua2luZy5ocGIuaHIvcmVzb3VyY2VzIiwiQXNzZWNvU0VF
LlBTRDIiXSwiY2xpZW50X2lkIjoiMjUudnJlYmFjLnRwcC1zdXBldXMiLCJzdWIiOiI0ODM4OGJjMi1iMWVmLT
Q5NjAtYTBiMC01MDk5Yzk0OWNhYzYiLCJhdXRoX3RpbWUiOjE2NDY5MjgzNzUsImlkcCI6ImxvY2FsIiwicHJv
dmlkZXIiOiIyNSIsImJhbmtfaWQiOiIxIiwiY3VzdG9tZXJfbnVtYmVyIjoiMzk1NTQ3NDM5MzEiLCJ1c2VyX2
5hbWUiOiIzOTU1NDc0MzkzMSIsInVzZXItdHlwZSI6InVzZXIiLCJwYXJ0eS1raW5kIjoiaW5kaXZpZHVhbCIs
ImUtYmFua2luZyI6ImUtYmFua2luZy1ub25lIiwiaXBfYWRkcmVzcyI6IjE3Mi4yNy4wLjk4Iiwic2NvcGUiOl
siQUlTOjljMmJkNmE5LTlkNzAtNDE5My04MDYzLWRmOTQyYjBiZGZlZCJdLCJhbXIiOlsicHdkIl19.X-
E9Oia7r2ds8b4NiRIYtZeBMkoMnqorTHlzGe9dgDENmK7Bh4SBjF4fxWiqgiXmdSFesgx-
SnHWgbL5C1v_A10WiQ9Ghh-lU0OjJAchX32YtRlQpaakux3Ew1PyTtCA-
xVHorc_2rI3o0Yd9aUSffUA4BqUxkT7sMHphMi_kyPRFxyuBvSFRTHHeF3D2Z3jDFKuVJVXnEsa6GjV1mv3_Ek
Fp5huq103fRt_2kpdY4d4w1gI1tm-
74idKwYRM2pVUZ0rTsNFFLlrkXPYUEAMUeJpzUd9MrooOTrJkhAy3YI2Ks8AEKydvljtLVMYJyRo1RRjsWzzfe
_Y60ZQaRuDVg",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

3.3.4. Finishing authorization

In order to finish payment resource authorization, TPP has to send a request to PUT */v1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}*. Payment id is id of payment resource that is authorizing, and *authorization id* is id of authorization resource that was created for payment resource authorization.

Request body of this field has to be in *application/json* format and must contain field *scaAuthenticationData*. Value of this field has to be equal to the access token that was obtained through the OAuth2 protocol.

Request body example:

```
{
  "scaAuthenticationData":
"eyJhbGciOiJSUzI1NiIsImtpZCI6IjU2OTJGQzc2RkI1Qjk4OTBCQUFGQ0JCQUYyOTQyNDZDQjUzMzdEMEYiL
CJ0eXAiOiJKV1QiLCJ4NXQiOiJWcEw4ZHZ0Ym1KQzZyOHU2OHBRa2JMVXpmUTgifQ.eyJuYmYiOjE2NDY5Mjgz
ODAsImV4cCI6MTY0NjkzMTk4MCwiaXNzIjoiaHR0cHM6Ly9pYW0udGVzdC5vcGVuYmFua2luZy5ocGIuaHIiLC
JhdWQiOlsiaHR0cHM6Ly9pYW0udGVzdC5vcGVuYmFua2luZy5ocGIuaHIvcmVzb3VyY2VzIiwiQXNzZWNvU0VF
LlBTRDIiXSwiY2xpZW50X2lkIjoiMjUudnJlYmFjLnRwcC1zdXBldXMiLCJzdWIiOiI0ODM4OGJjMi1iMWVmLT
Q5NjAtYTBiMC01MDk5Yzk0OWNhYzYiLCJhdXRoX3RpbWUiOjE2NDY5MjgzNzUsImlkcCI6ImxvY2FsIiwicHJv
dmlkZXIiOiIyNSIsImJhbmtfaWQiOiIxIiwiY3VzdG9tZXJfbnVtYmVyIjoiMzk1NTQ3NDM5MzEiLCJ1c2VyX2
5hbWUiOiIzOTU1NDc0MzkzMSIsInVzZXItdHlwZSI6InVzZXIiLCJwYXJ0eS1raW5kIjoiaW5kaXZpZHVhbCIs
ImUtYmFua2luZyI6ImUtYmFua2luZy1ub25lIiwiaXBfYWRkcmVzcyI6IjE3Mi4yNy4wLjk4Iiwic2NvcGUiOl
siQUlTOjljMmJkNmE5LTlkNzAtNDE5My04MDYzLWRmOTQyYjBiZGZlZCJdLCJhbXIiOlsicHdkIl19.X-
E9Oia7r2ds8b4NiRIYtZeBMkoMnqorTHlzGe9dgDENmK7Bh4SBjF4fxWiqgiXmdSFesgx-
SnHWgbL5C1v_A10WiQ9Ghh-lU0OjJAchX32YtRlQpaakux3Ew1PyTtCA-
xVHorc_2rI3o0Yd9aUSffUA4BqUxkT7sMHphMi_kyPRFxyuBvSFRTHHeF3D2Z3jDFKuVJVXnEsa6GjV1mv3_Ek
Fp5huq103fRt_2kpdY4d4w1gI1tm-
74idKwYRM2pVUZ0rTsNFFLlrkXPYUEAMUeJpzUd9MrooOTrJkhAy3YI2Ks8AEKydvljtLVMYJyRo1RRjsWzzfe
_Y60ZQaRuDVg"
}
```

4. AIS and PIS resources using the Decoupled flow

4.1. Description and Preconditions

Goal of this section is to describe how to use decoupled flow for authentication and authorization.

Decoupled flow of authentication extends available methods for TPP and PSU. In order to use this approach, there are several conditions that need to be met:

- PSU needs to have active mobile banking app service or active mobile token service – all available in bank's single mobile application named mHPB available on Android, iOS and Huawei OS,
- PSU mobile device must not be disabled for push notifications from mHPB.

When using DECOUPLED flow as SCA approach, PSU will receive push notification within mHPB mobile banking application that will allow user to perform a secure login, have overview of TPP data for related consent or payment and perform SCA.

It is important to emphasize that Decoupled flow fully completes/authorizes AIS or PIS resource and therefore, TPP does not need to send any additional requests to our API (like accessToken within PUT request Redirect flow).

4.2. Creating resources

To create AIS or PIS resource, use request body as usual.

In order to use this method of authentication, TPP should add extra headers to the POST request when creating resource, specified here:

- TPP-Decoupled-Preferred: **true** (*mandatory*; *or any other proper combination from the table below*)
- TPP-Redirect-URI: https://example.com/OK *(optional),*
- TPP-Nok-Redirect-URI: https://example.com/NOK *(optional).*

HEADERS		
TPP-Redirect-Preferred	TPP-Decoupled-Preferred	INVOKED SCA APPROACH
not present	TRUE	DECOUPLED
not present	FALSE	REDIRECT
TRUE	*not present*	REDIRECT
FALSE	*not present*	DECOUPLED
FALSE	TRUE	DECOUPLED
TRUE	FALSE	REDIRECT
TRUE	TRUE	DECOUPLED
FALSE	FALSE	REDIRECT
not present	*not present*	REDIRECT

List of possible combinations and the resulting system behavior.

4.3. Starting authorization

To start authorization on AIS or PIS resource, use request body and request headers as usual. If everything is OK, you will receive statusCode=201.

In case of response **statusCode=201** and element **“scaStatus”=”psuIdentified”** in response body, PSU will instantly receive push notification on the mobile device that can be opened and authorized with SCA available on mobile banking app.

In case of any other statusCode or any error in response body, TPP should consider that this PSU is not capable of receiving push notifications and should choose another method for authorization (oAuth Redirect flow) by creating new resource and initiating new authorization.

Response example:

```
{
    "scaStatus": "psuIdentified",
    "authorizationId": "d4efd448500f44378340057418d578f9",
    "scaMethods": null,
    "_links": {
        "updatePsuAuthentication": {
            "href": "v1/payments/domestic-credit-transfers-
hr/6f3a688f71274643a799aed6c2dfa227/authorisations/d4efd448500f44378340057418d578f9"
        },
        "scaStatus": {
            "href": "v1/payments/domestic-credit-transfers-
hr/6f3a688f71274643a799aed6c2dfa227/authorisations/d4efd448500f44378340057418d578f9"
        }
    }
}
```

5. AIS resource implementation guidelines

Goal of this section is to help TPPs to have a seamless implementation with our PSD2 API and according to Berlin Group, RTS and Croatian national framework for PSD2.

5.1. Parameter "frequencyPerDay"

This parameter determines for how many times during a day (during a 24-hour timeframe; 00:00-23:59) TPP can make requests for a given consent, without PSU presence. The value must be in range of 1 to 4 and is set and validated during a creation of AIS resource.

HPB supports this parameter as described: If consent is valid and request is made without PSU presence, HPB will group all requests made by TPP in given timeframe and count them as one request. The length of the timeframe is set to 4 minutes.

Example: If „frequencyPerDay" parameter is set to '4', it will allow TPP to have four timeframes daily to make multiple number of requests and all requests within each timeframe will be counted as one.

Requests that are made with valid "PSU-IP-Address" header are considered to be actively initiated by the PSU and therefore are not counted against „frequencyPerDay" parameter.

5.2. Transactions history

As per Delegation on RTS, TPP is allowed to access PSU transactions history for a maximum of 90 days in past. To access history older than 90 days, PSU must perform SCA every time.

HPB supports this feature as described: After PSU successfully authorizes AIS consent, TPP will be granted permission to access unlimited transactions history of PSU (Note: not older/further than on other online and mobile banking channels), but only during a first request to the endpoint: GET v1/accounts/{{resourceId}}/transactions

To make a successful request and access transactions history, TPP should be aware that query parameters for *bookingStatus*[3] and *dateFrom*[4] must always be provided. Query parameter *dateTo*[5] is optional, and if not provided, it will be considered as *currentDate*.

Any other request for transactions history (apart first one ever during a consent validity) will be checked against 90-days rule.

[3] Supported query parameters: booked. Parameters *information/pending/both/all* are not supported currently and will always return an empty array.

[4] Must be in ISO format: YYYY-MM-DD

[5] Must be in ISO format: YYYY-MM-DD and cannot be in the future OR older than *dateFrom.*

Transaction history response body will contain data (elements) that is equal to other online and mobile banking channels in HPB (including debtor/creditor element). If TPP needs to access more detailed information for particular transaction from the transaction list, TPP is advised to make transaction details request using transaction unique identifier.

5.3. Transactions Response Pagination

When accessing transactions history for a given account and given period, HPB will assess a total count of transactions that must be presented to TPP and in case that total count exceeds 25, pagination will be invoked.

HPB supports pagination of transactions response as described:

- Pagination is invoked only if more than 25 transactions need to be presented,
- If pagination is invoked, response body will contain new element with a „next“ link,
- Links to „previous“, „first“ and „last“ page are not supported on Production environment,
- Every page within the invoked pagination contains 25 transactions (last one possibly less),
- Last page of pagination will not have „next“ link signalizing to be the last page,
- TPP should make GET request to every „next“ link to access all transactions,
- TPP should keep request headers as in initial call,
- The „next“ link given consists of query parameters that must not be changed,
- Pagination resource (all pages) is available for 3 minutes and expires after (for every page).

TPP should be aware that some business users (*large corporate clients*) might have a huge number of transactions and should therefore optimize their query by using *dateFrom* and *dateTo* query parameters to build more requests with smaller periods.

Response example:

```
"_links": {
            "next": {
                "href": "/v1/accounts/9000197900/transactions?bookingStatus=booked&dateFrom=2018-01-01&dateTo=2022-03-01&tranId=3456abc&withBalance=false"
                     }
              }
```

6. PIS resource implementation guidelines

6.1. Without debtor account

Additional possibility for TPP and PSU clients is creating a payment resource without providing *debtorAccount* object in JSON. In that case, PSU will have option to select any IBAN it has access to, after completing SCA.

This option helps TPP clients to have a more flexible payment initiation and helps PSU to select any IBAN available as debtor account, during the authorization of the payment. This feature is available both on oAuth2 Redirect and Decoupled approach.

```
{
  "instructedAmount": {
    "currency": "HRK",
    "amount": "1.99"
  },
  "creditorAccount": {
    "iban": "HR9323900010000000410"
  },
  "creditorName": "John Doe",
  "creditorAddress": {
        "streetName": "Test street",
        "buildingNumber": "1",
        "townName": "ZAGREB",
        "postCode": "10000",
        "country": "HR"
   },
  "purposeCode": "OTHR",
  "remittanceInformationUnstructured": "Free Description Here",
  "requestedExecutionDate": "YYYY-MM-DD",
}
```

7. Confirmation of Funds/Funds Check Service (CoF/FCS)

To start the request, only TPPs with a valid PIISP license should make a request to */v1/funds-confirmations* endpoint with body shown in example below:

```
{
"account":
    {"iban": "HR9323900010000000410"},
        "instructedAmount": {
            "amount": "1.00",
            "currency": "HRK"
}}
```

Response will contain element “fundsAvailable” with value “true” or “false” depending if there is enough amount of money, or not.

Please note that HPB supports only requests for IBAN accounts. Card accounts requests are not supported.

8. Support

Email: psd2.support@hpb.hr (*Support in Croatian and English.)*

NOTE: When creating support request or sending email enquiry for the Production environment, TPP should follow guidelines and rules defined in our “Support procedure for problems and incidents related to the Bank’s dedicated interface” document that regulates reporting procedure, resolution procedure, response and resolution time, working hours and escalation process (*available on the link below*).

Useful links & Related documents:

HPB PSD2 dedicated webpage: https://www.hpb.hr/hr/psd2-hpb-open-api-portal/318

HPB PSD2 API docs: https://api.openbanking.hpb.hr